## 简介

用一个对象封装其他对象的相互交互，中介者是各对象不需要显示的相互引用，从而实现其松耦合，从而可以独立地改变他们之间的交互。

## 角色

1. 抽象中介者（mediator）：各同事对象的通信接口。
2. 具体中介者（ConcreteMediator）：协调各同事对象实现写作行为，了解并维护同事。
3. 抽象同事类（colleague）：同事类的抽象接口
4. 具体同事类（ConcreteColleague）：拥有各自行为的同事，通过中介者通讯。

## 优缺点

优点：

1. 降低耦合，是同事类之间可以相互独立使用
2. 降低复杂度，利于理解和维护同事类之间的关系。
3. 将对象的行为和协作进行抽象，能够比较灵活的处理对象之间的相互作用。

## 使用场景

一般来说，同事之间的关系时复杂的，多个同事之间相互关联时，他们之间的关系会形成复杂的网状结构，这会导致过度耦合，即不利于类的复用，也不稳当。

如果引入中介者模式，同事之间的关系会形成形状结构，任何一个类的变动，自会影响它本身和中介者，这样就减少了类的耦合。

一个好的设计，必定不会把所有对象之间关系的处理逻辑封装在本类中，而是使用一个专门的类来管理那些不属于自己的行为。

## 参考

<http://blog.csdn.net/jason0539/article/details/45216585>